

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

Copy 830304--14

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

LA-UR-83-29

U.S. GOVERNMENT


TITLE: INTERCOMPARISON OF THE FINITE DIFFERENCE AND NODAL DISCRETE ORDINATES
AND SURFACE FLUX TRANSPORT METHODS FOR A LWR POOL-REACTOR BENCHMARK
PROBLEM IN X-Y GEOMETRY

AUTHOR(S): R. D. O'Dell
J. Stepanek
M. R. Wagner

SUBMITTED TO: American Nuclear Society
Mathematics and Computation Division Topical Meeting
"Advances in Reactor Computations"
Salt Lake City, Utah
March 28-31, 1983

NOTICE

PORTIONS OF THIS REPORT / It
has been reproduced for / to
copy to permit the broad / avail-
ability.

By acceptance of this article  publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce
the published form of this contribution, or to allow others to do so, for U.S. Government purposes

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

MASTER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

INTERCOMPARISON OF THE FINITE DIFFERENCE AND NODAL DISCRETE ORDINATES
AND SURFACE FLUX TRANSPORT METHODS FOR A LWR POOL REACTOR BENCHMARK
PROBLEM IN X-Y GEOMETRY

R.D. O'Dell
University of California
Los Alamos National Laboratory,
Los Alamos, Mexico 87545, USA

J. Stepanek
Swiss Federal Institute for Reactor Research,
5303 Wuerenlingen, Switzerland

and

M.R. Wagner
Kraftwerk Union AG
8520 Erlangen, Fed. Rep. of Germany

The aim of the present work is to compare and discuss the three of the most advanced two dimensional transport methods, the finite difference and nodal discrete ordinates and surface flux method, incorporated into the transport codes TWODANT, TWOTRAN-NODAL, MULTIMEDIUM and SURCU.

For intercomparison the eigenvalue and the neutron flux distribution are calculated using these codes in the LWR pool reactor benchmark problem. Additionally the results are compared with some results obtained by French collision probability transport codes MARSHAS and TRIDENT. Because the transport solution of this benchmark problem is close to its diffusion solution some results obtained by the finite element diffusion code FINELM and the finite difference diffusion code DIFF-2D are included.

INTERCOMPARISON OF THE FINITE DIFFERENCE AND NODAL DISCRETE ORDINATES
AND SURFACE FLUX TRANSPORT METHODS FOR A LWR POOL REACTOR BENCHMARK
PROBLEM IN X-Y GEOMETRY

INTRODUCTION

In spite of the great deal of progress made in recent years in transport theory methods the performance of these is still far from being satisfactory. Particularly objectionable are the long computing times encountered when the number of variables, spatial meshes and energy groups becomes very large in 2 and 3 dimensions.

The most widely used method is the discrete ordinates method (S_N theory). This method is based on the solution of the integro-differential equation using discrete angular integration. In 2 and 3 dimensions it also leads to a simpler formalism than other methods.

However, its main disadvantage is the so-called ray effect.

In addition the use of the finite difference scheme in space with just flat or linear flux approximation requires in many problems up to tens of thousands of meshes for the adequate accuracy of the solution in two dimensions.

In many problems especially for large number of unknowns the inner iterative schemes are often inefficient and lead to a very large number of iterations and therefore to high computational time.

In recent years a great deal of time and effort has been therefore devoted to mitigating the ray effect and to accelerating the convergence of inner iteration schemes by introducing the new powerful acceleration and coarse-mesh rebalancing methods.

Different S_N spatial differencing approximations have also been introduced.

The need for developing coarse-mesh transport theory methods which allow one to dramatically decrease the number of spatial meshes in core design, fusion blanket and shielding analysis for configurations containing large homogeneous regions has lead to the development of different theories. These are generally based on nodal methods or on the surface flux transport method which uses the general order of Legendre polynomial series to approximate the spatial as well as angular flux distributions.

The aim of the present paper is to compare three of the most advanced two-dimensional transport methods, the finite difference and nodal discrete ordinates and surface flux method, calculating the same benchmark problem and investigating by this way the efficiency of different approaches, such

as surface flux approximations, boundary flux and volume spatial flux approximations.

The three above mentioned methods are incorporated into the finite difference S_N code TWODANT¹, the nodal S_N code TWOTRAN-NODAL^{2,3}, a nodal S_N code MULTIMEDIUM⁴ and the surface flux codes SURCU^{5,6,7}. For the sake of comparison some results by Y. Mederbel and Z. Stankovski (CEA, France) using the French collision probability codes MARSYAS⁸ and TRIDENT⁹ are also included.

An LWR pool reactor benchmark problem¹⁰ with fission sources is used for the calculations. This case is very difficult to calculate because the zones are optically thick (large value of $\Sigma_t \times$ dimension) and weakly connected. It is therefore difficult to obtain the results with high accuracy. Additionally the spatial flux distribution varies rapidly so that a very large number of meshes has to be used if the problem should be converged using the finite difference scheme only. This benchmark problem is therefore suitable to study the effectiveness of the methods which use a higher order of spatial flux approximation, such as the nodal approach of expansion of the spatial flux into Legendre polynomial series or approximating it using the higher order of Lagrange polynomials, for example.

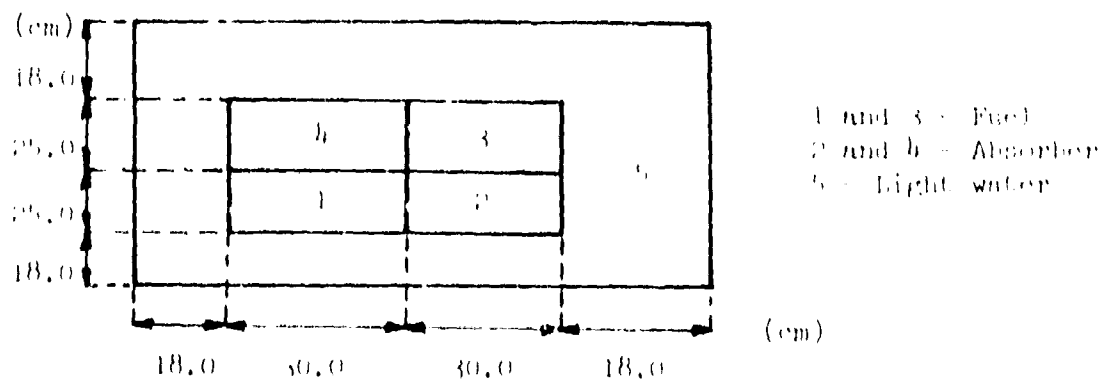
In contrast to the above, the order of S_N or P_N approximation of the surface fluxes is not very important.

Because the transport solution to this problem is close to its diffusion solution, the transport results are also compared with those obtained using the finite element diffusion code FINELM¹¹ and the finite difference diffusion code DIFF-2D (2DB)¹².

BENCHMARK PROBLEM

The second part of the benchmark problem "LWR pool reactor" defined by J. Stepánek for the IAEA coordinated research program on "Transport theory and advanced reactor calculations"¹⁰ was used for comparison in this paper. It consists of two large source zones and two large absorber zones surrounded by light water. One energy group, isotropic neutron scattering, and vacuum boundary conditions are considered. The axial buckling is zero.

Geometry



Neutron cross sections

Material	Σ_a	$\nu\Sigma_f$	Σ_t	Σ_{s0}	Σ_{s1}
1	0.07	0.079	0.60	0.53	0.27
2	0.28	0.0	0.48	0.20	0.02
3	0.04	0.043	0.70	0.66	0.30
4	0.15	0.0	0.65	0.50	0.15
5	0.01	0.0	0.90	0.89	0.40

Normalization: Total production = 1 (n/sec).

Eigenvalue and average fluxes in the material zones are calculated.

DEVELOPMENT OF THE DIAMOND-DIFFERENCE EQUATIONS

The widely-used diamond difference scheme (used in the diffusion synthetic accelerator TWODANT code at Los Alamos)¹ is readily developed by considering the i, j, g cell to be defined as the rectangle defined by $x_L \leq x \leq x_R$, $y_B \leq y \leq y_T$ and with $\Delta x = x_R - x_L$, $\Delta y = y_T - y_B$. The discrete-ordinates form of the transport equation for this cell for energy group g and direction m is

$$\mu_m \partial \psi_{m,g}(x,y) / \partial x + \eta_m \partial \psi_{m,g}(x,y) / \partial y + \sigma_g \psi_{m,g}(x,y) = S_{m,g}(x,y) . \quad (1)$$

Hereafter the subscript m and g will be omitted.

If we integrate Eq. (1) over the i, j, g cell we obtain

$$\mu(\psi_R - \psi_L) \Delta y + \eta(\psi_T - \psi_B) \Delta x + \psi_{av} \sigma \Delta x \Delta y = S_{av} \Delta x \Delta y \quad (2)$$

where ψ_R is the average angular flux over the right face of the cell, ψ_L is the average flux over the left face of the cell, ψ_T and ψ_B are the average fluxes over the top and bottom cell faces, respectively, ψ_{av} is the average angular flux in the cell, and S_{av} is the cell-average source.

We now assume μ and η are positive, without loss of generality, so that ψ_L and ψ_B may be assumed known thus leaving the three unknowns ψ_R , ψ_L , and ψ_{av} to be determined. The remaining two equations (in addition to Eq. (2)) required for closure are the "diamond-difference" relations

$$\psi_R = 2\psi_{av} - \psi_L \quad (3)$$

and

$$\psi_T = 2\psi_{av} - \psi_B . \quad (4)$$

The solution procedure is to simply substitute Eqs. (3) and (4) into Eq. (2) and then solve for ψ_{av} . Then the outflow fluxes ψ_R and ψ_T are computed from the diamond-difference relations. It is readily seen that the outflow fluxes, when extrapolated using Eqs. (3) and (4), can be negative. When this occurs the offending negative outflow flux is set to zero and Eq. (2) is again solved (with that outflow flux explicitly set to zero) for ψ_{av} . This procedure is called the "set-to-zero fixup". The use of fixups causes some degree of uncertainty in the accuracy of the computed results, however. In addition, flux fixups tend to interact rather unfavorably with iteration acceleration schemes, that is, acceleration schemes are generally less effective when many flux fixups are needed. To reduce the number of flux fixups requires reducing the mesh cell sizes (increasing the number of mesh cells), an option that increases both the computer storage requirements and the execution time. In many multidimensional problems it is simply not possible to refine the mesh sufficiently. The negative flux/flux fixup problem with the diamond difference scheme is equivalent to saying that the scheme is an insufficiently accurate spatial differencing scheme for many multidimensional transport problems. This is seen in the table of results for the benchmark problem analyzed in this paper using the diamond-difference code TWODANT. Even at the 120 x 120 mesh the problem is still not fully converged to the "exact" k_{eff} and zone-averaged fluxes for the S_8 quadrature used.

Thus, even though diamond-difference codes such as TWOTRAN-II² and DOT-4¹³ are still the most commonly used deterministic transport codes, there is a great need for more production codes employing more accurate differencing scheme without an undue increase in computational time or computer storage requirements.

DEVELOPMENT OF THE LOS ALAMOS NODAL EQUATIONS

The Los Alamos transport nodal methods can be characterized by the separate angular flux expansions assumed on the edges of a node (cell) and over the interior of the node. For example, a "constant-linear" method refers to one in which independent constant edge angular flux "expansions" are assumed together with a linear flux expansion over the node interior. The nodal method used for this paper is the "linear-linear" method defined as method LL4 in Ref. 3.

Consider the i,j node to be the rectangle defined by $x_L \leq x \leq x_R$, $y_B \leq y \leq y_T$ and with $\Delta x = x_R - x_L$, $\Delta y = y_T - y_B$, $x_i = (x_L + x_R)/2$ and $y_j = (y_B + y_T)/2$. The discrete-ordinates equation for direction m and energy group g is

$$\mu_m \partial \psi_{m,g}(x,y) / \partial x + \eta_m \partial \psi_{m,g}(x,y) / \partial y + \sigma_{t,m,g} \psi_{m,g}(x,y) = S_{m,g}(x,y) \quad (5)$$

Hereafter the subscripts m and g will be dropped.

We assume that the source within the node is linear, that is,

$$S(x,y) = S_{av} + S_x * 2(x - x_i) / \Delta x + S_y * 2(y - y_j) / \Delta y \quad (6)$$

and that the source for a given iteration is determined using fluxes from the previous iteration. It is further assumed, for this analysis, that $\mu > 0$, $\eta > 0$ so that the fluxes on the node edges at x_L and y_B are known.

We represent the angular flux along the top edge of the node by the linear expansion:

$$\Psi(x, y_T) = \psi_T + \theta_T * 2(x - x_i)/\Delta x \quad \begin{cases} x_L < x < x_R \\ y = y_T \end{cases} \quad (7)$$

with similar expressions for the other node edges.

Nodal equations are generated by performing transverse integrations of (5) over the node in the x- and y-directions. If we integrate (5) over $\{x_L, x_R\}$ there results an ordinary differential equation for $\Psi^0(y) \equiv$

$$\frac{1}{\Delta x} \int_{x_L}^{x_R} \Psi(x, y) dx :$$

$$\eta d\Psi^0(y)/dy + \sigma \Psi^0(y) = S_{av} + S_y * 2(y - y_j)/\Delta y - (\mu/\Delta x) [\Psi(x_R, y) - \Psi(x_L, y)] \quad (8)$$

We can also multiply (5) through by $(6/\Delta x^2)(x - x_i)$ and x-integrate over the node to get an equation for the y-dependent, first flux moment $\Psi^1(y) \equiv$

$$6/\Delta x^2 * \int_{x_L}^{x_R} (x - x_i) \Psi(x, y) dx :$$

$$\eta d\Psi^1(y)/dy + \sigma \Psi^1(y) = S_x + (6\mu/\Delta x) * \left\{ \Psi^0(y) - 0.5 [\Psi(x_R, y) + \Psi(x_L, y)] \right\} \quad (9)$$

Similar expression for $\Psi^0(x)$ and $\Psi^1(x)$ can be obtained by integrating in the y-direction.

Exact solutions to (8) and (9) are obtained and evaluated at $y = y_T$. These are:

$$\Psi^0(y_T) = \psi_T = \psi_B \exp(-c_y) + (1/\eta) * \int_{y_B}^{y_T} dy' \exp(-\sigma (y_T - y')/\eta) * \quad (10)$$

$$\left\{ S_{av} + (2/\Delta y)(y' - y_j) * S_y - (\mu/\Delta x) \left[\psi(x_R, y') - \psi(x_L, y') \right] \right\} \quad (10)$$

and

$$\psi^1(y_T) = \theta_T = \theta_B \exp(-\epsilon_y) + (1/\eta) * \int_{y_B}^{y_T} dy' \exp\{-\sigma(y_T - y')/\eta\} * \quad (11)$$

$$\left\{ S_x + (6\mu/\Delta x) \left[\psi^0(y') - 0.5\{\psi(x_R, y') + \psi(x_L, y')\} \right] \right\} .$$

Similar exact expressions for $\psi^0(x_R) \equiv \psi_R$ and $\psi^1(x_R) \equiv \theta_R$ are also generated. In these expressions we have defined $\epsilon_y \equiv \sigma\Delta y/\eta$ and $\epsilon_x \equiv \sigma\Delta x/\mu$.

For the linear-linear nodal method it is assumed that the interior flux $\psi(x, y)$, $x_L < x < x_R$, $y_B < y < y_T$, is given by

$$\psi(x, y) = \psi_{av} + \psi_x * 2(x - x_i)/\Delta x + \psi_y * 2(y - y_j)/\Delta y \quad (12)$$

Equation (6) is used for computing the linear source distribution within the cell.

For the seven unknowns ψ_T , ψ_R , ψ_{av} , ψ_x , ψ_y , θ_T , and θ_R we require seven equations. Three of these equations are conservation equations found by taking the 1, $(6/\Delta x^2) * (x - x_i)$, and $(6/\Delta y^2)(y - y_j)$ moments of Eq. (5).

The remaining four equations are found by inserting the linear flux expansions directly into Equations (10) and (11) and their companion equations for $\psi^0(x_R)$ and $\psi^1(x_R)$ and analytically performing the integrations.

The manner in which these seven equations are solved requires careful treatment to ensure a proper solution when σ is small or zero.

This nodal method was programmed into a special version of the TWOTRAN-II² code at Los Alamos. No attempt was made to provide this nodal code with an effective iteration acceleration scheme. Accordingly, the nodal computer runs converged very slowly. Before the linear nodal method at Los Alamos can achieve production code status, an effective iteration acceleration must be devised and implemented. Work in this remains to be performed.

Generally, the Los Alamos linear nodal method described in this paper and in Reference 3, requires about 2.5 times as much computational time as the diamond difference scheme per mesh cell calculation. Additionally, the linear nodal method requires about twice as much computer storage as does a comparable diamond difference code. The linear nodal method, however, generally produces results of comparable accuracy to those from diamond-differencing but with far fewer mesh cells, especially if one is interested

in pointwise quantities. A net savings in both computer time and storage is obtainable using the linear-linear nodal scheme when compared with the diamond-difference scheme for the same accuracy.

NODAL DISCRETE ORDINATES METHOD AT KWU

The nodal discrete ordinates method⁴ (NDOM) is a hybrid method for the approximate solution of the two-dimensional transport equation in rectangular x,y geometry. The method combines certain features of the integral transport theory and the discrete ordinates method with ideas derived from the nodal diffusion theory approach^{4,14}. As in the latter, the primary variables are the node average scalar fluxes and the surface averaged partial currents J_{\pm}^+ at the node boundaries. To simplify the notation, the group index g will^g be omitted hereafter. The angular distribution of the incoming and outgoing surface fluxes is described in terms of average half range angular flux moments defined by

$$\phi^{+(n)}(u_s) = \frac{1}{2\pi\Delta v} \int_0^{\Delta v} dv \int_0^1 u^n d\mu \int_0^{2\pi} d\phi_u \psi(u,v,\mu,\phi_u), \quad (13)$$

where u is the spatial direction normal to the node boundary at $u=u_s$; μ is the direction cosine and ϕ_u is the azimuthal angle with respect to the u -axis. According to (13) these angular flux moments are defined by formally integrating over both the transverse spatial and angular dimensions (direction v and azimuthal angle ϕ_u) of the node. The actual nodal variables of NDOM are the partial currents on the left (l) and right (r) node boundaries

$$J_{\pm}^+(u_s) = \phi_{\pm}^{+(1)}(u_s), \quad \begin{matrix} u = x, y \\ s = l, r \end{matrix}, \quad (14)$$

and the average half range μ -moments (with $n=1, \dots, n_{\text{Max}}$)

$$\mu_{\pm}^{+(n)}(u_s) = \frac{\phi_{\pm}^{+(n)}(u_s)}{\phi_{\pm}^{+(0)}(u_s)}. \quad (15)$$

Note, that from these variables the surface averaged scalar fluxes on the node edges are determined as

$$\phi(u_s) = \frac{J^+(u_s)}{\mu^{+(1)}(u_s)} + \frac{J^-(u_s)}{\mu^{-(1)}(u_s)} \quad (16)$$

In the NDOM version as currently implemented in the nodal multigroup program MULTIMEDIUM n_{Max} is equal to 2, which corresponds to assuming a DP_2 approximation for the angular distribution of the surface fluxes.

In analogy to the procedure in nodal diffusion theory methods the basic iterative step in solving the NDOM equations is to compute, for a fixed set of in-currents and in-moments on all faces of a particular node, a new set of corresponding out-currents and out-moments by solving a system of coupled one-dimensional transverse integrated transport equations⁴. These auxiliary equations are derived by applying the operator

$$\frac{1}{2\pi\Delta v} \int_0^{\Delta v} dv \int_0^{2\pi} d\phi_\mu$$

to the two-dimensional transport equation with isotropic scattering. As a result, a system of two one-dimensional slab type equations is obtained

$$\mu \frac{\partial \Psi(u, \mu)}{\partial u} - \sigma_t \Psi(u, \mu) - \sigma_s \Phi(u) = Q(u) + L^V(u, \mu), \quad (17)$$

with $u=(x,y)$, for the "double" transverse integrated angular fluxes $\Psi(u, \mu)$. Both equations (17) are coupled through the transverse leakage term $L^V(u, \mu)$. In the simplest version of NDOM⁴ the transverse leakage is assumed to be isotropic and spatially constant. Hence, $L^V(u, \mu)$ is expressed in terms of the partial currents in the transverse direction v

$$L^V(u, \mu) = \frac{1}{\Delta v} \left[J^{+v}(\Delta v) - J^{-v}(\Delta v) + J^-(0) - J^+(0) \right]. \quad (18)$$

The discrete ordinate form of Eq. (17) is discretized spatially as in conventional S_N -codes, using the diamond differencing scheme for an equidistant spatial mesh. These equations are solved, for each energy group, by a direct inversion of the streaming-collision operator. This then obviates the need for iterating on the scattering source and for storing the one-dimensional group sources from one global outer iteration to the next. As a result, the same overall iteration strategy and the same methods of convergence acceleration can be used as for nodal diffusion theory calculations¹⁴.

The nodal discrete ordinates method has been implemented in the framework of the nodal code MULTIMEDIUM by, essentially, adding a central subroutine and by assigning additional storage fields for the new variables $\mu^\pm(n)$. In this form MULTIMEDIUM is predominantly used for carrying out transport-depletion calculations for the analysis of heterogeneous LWR fuel assemblies^{15,16}. Benchmark calculations¹⁵ for a number of PWR and BWR assembly problems have demonstrated the accuracy and computational efficiency of the nodal discrete ordinates method for this type of application.

The results presented in this paper show that NDOM allows one also to solve transport theory problems involving large homogeneous regions with reasonable accuracy and short computing times. Another example is the ZPPR-7A problem for which some results are given in reference 16. From the results listed in Table 2 for the EIR-3B problem one notes that the NDOM solutions do not exactly converge to the asymptotic solution of the explicit two-dimensional S_N codes TWODANT or TWOTRAN-NODAL. There are several reasons for this behavior. One is the fact that a double Gauss quadrature has been used in the one-dimensional nodal S_8 calculations. However, it is believed

that the discrepancy is mainly a consequence of the simplifying assumption of isotropic transverse leakage, Eq. (18). For the present problem, as well as for many other problems studied, the resulting error is not large. Therefore, it was decided to accept this error, for the time being, and to put the emphasis on consolidating the isotropic leakage version of NDOM in MULTIMEDIUM for routine use in design oriented LWR applications. On the other hand, preliminary investigations indicate that it is possible to correct for this error by introducing suitable angular (and spatial) approximations of the transverse leakage term $L^V(u, \mu)$ in Eq. (17). The development of such an improved and stable algorithm remains an objective of future work.

SURFACE FLUX MULTIPLE P_N METHOD

The general surface flux method is used to solve the integral neutron transport equation in X-Y geometry. The spatial distribution of neutron flux and sources is represented by a Legendre polynomial expansion while zonal spherical harmonics, Y^Z , orthogonal in angular zones, are used to approximate the incoming and outgoing surface fluxes in I_z angular zones at the interfaces between space intervals.

This method is incorporated into the code SURCU⁵. SURCU is also written for one-dimensional geometries, plane, spherical, and cylindrical, considering the double P_N approximation of the angular fluxes at the mesh boundaries. In X-Y geometry it subdivides space into rectangular meshes and solves the integral equation in each mesh. Meshes are then connected to each other through incoming surface currents on the mesh boundaries. The angular space at the mesh boundaries is first subdivided into four angular quadrants, each of which is then further divided into I_z angular zones. In each angular zone zonal spherical harmonics (i.e. angular functions orthogonal in a zone) are used to develop the angular distribution of surface fluxes. The spatial distribution of the neutron flux and of the sources is represented by a Legendre polynomial expansion.

In the present work all calculations were performed with $I_z = 1$, i.e. the angular space at the mesh boundaries was subdivided into four quadrants only.

Consider a geometrical configuration consisting of a number of homogeneous rectangular meshes. Then the integral transport equation for the neutron flux $\phi(\bar{r}, \bar{\Omega})$ for energy group g at \bar{r} within a single mesh (i, j) takes the form

$$\phi(\bar{r}, \bar{\Omega}) = \int_0^s ds' q(\bar{r}', \bar{\Omega}) \frac{e^{-\frac{\Sigma(\bar{r}-\bar{r}')}{\sin\theta}}}{\sin\theta} + \phi(\bar{r}_B, \bar{\Omega}) e^{-\frac{\Sigma s}{\sin\theta}}, \quad (19)$$

where $\phi(\bar{r}, \bar{\Omega})$, $q(\bar{r}, \bar{\Omega})$ and $\phi(\bar{r}_B, \bar{\Omega})$ are angular neutron flux, total angular volume source and angular neutron boundary flux, respectively, Σ is the total macroscopic cross section, the direction $\bar{\Omega}$ is defined by the axial and azimuthal angles θ and ϕ , respectively, \bar{r}_B is a point at the surface of

the mesh (i,j), and \bar{r}' is a point on the line connecting \bar{r} and \bar{r}_B .

In the above formula and hereafter the subscript g denoting the energy group g will be omitted. As usual, one now assumes that the angular dependence of the flux ϕ , the total volume source q, and the given volume source S can be expanded into spherical harmonics:

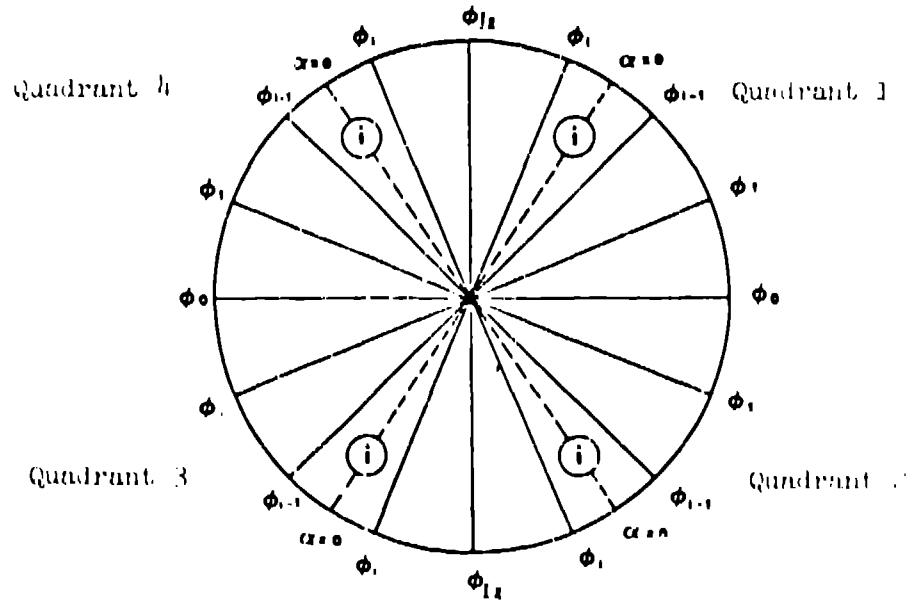
$$\begin{pmatrix} \phi(\bar{r}, \bar{\Omega}) \\ q(\bar{r}, \bar{\Omega}) \\ S(\bar{r}, \bar{\Omega}) \end{pmatrix} = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \sum_{m=0}^{\ell} \sum_{k=0}^{\text{Min}(m,1)} Y_{\ell mk}(\bar{c}, \bar{\phi}) \begin{pmatrix} \phi_{\ell mk}(\bar{r}) \\ q_{\ell mk}(\bar{r}) \\ S_{\ell mk}(\bar{r}) \end{pmatrix}, \quad (20)$$

for $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. Owing to the axial symmetry of fluxes and sources in X-Y geometry only each spherical harmonics appear in Eq. (20), $\ell+m = \text{even}$. The angular volume source moments q are defined by

$$q_{\ell mk}(\bar{r}) = S_{\ell mk}(\bar{r}) + \Sigma_{S\ell} \phi_{\ell mk}(\bar{r}), \quad (21)$$

where $\Sigma_{S\ell}$ is the ℓ -th component of the macroscopic scattering cross section and the given volume source moments $S_{\ell mk}$ include the contributions from other energy groups.

On each side of mesh (i,j) one now subdivides the angular space formed by the flight directions into four quadrants and each quadrant into 1_z zones.



Then the boundary angular flux is expanded in each zone into a series of zonal spherical harmonics $Y_{\ell mk}^z(0, \phi, 0, \phi_z)$:

$$\phi(\bar{r}_B, 0, \phi) = \sum_{\ell=0}^N \frac{(2\ell+1)}{4\pi \phi_{z\ell}} \sum_{m=0}^{\ell} \sum_{k=0}^{\text{Min}(m,1)} Y_{\ell mk}^z(0, \alpha, 0, \phi_{zn}) \phi_{\ell mk}^{zn}(\bar{r}_B), \quad (22)$$

for $\ell+m$ even, $0 \leq \theta \leq \pi$, $\phi_{n-1} \leq \alpha \leq \phi_n$, and $-\phi_{zn} \leq \alpha \leq \phi_{zn}$. The angle α is measured from the symmetry axis of the angular zone. ϕ_{zn} is equal to $(\phi_n - \phi_{n-1})/2$, where ϕ_n and ϕ_{n-1} are boundary angles of the n 'th zone starting at the x-axis with $n=0$ (see Fig. on the previous page). N is the order of the P_N approximation. $\ell+m$ is again chosen to be even because the angular flux is an even function of θ in X-Y geometry.

The boundary multiple P_N flux moments $\phi_{\ell mk}$ are given by the equation

$$\phi_{\ell mk}^{xn}(\bar{r}_B) = \int_{-\phi_{zn}}^{\phi_{zn}} d\alpha \int_0^\pi d\theta \sin\theta Y_{\ell mk}^Z(\theta, \alpha, 0, \phi_{zn}) \phi(\bar{r}_B, \theta, \alpha). \quad (23)$$

The zonal spherical harmonics Y^Z are defined and their calculational method is described in Ref. 6. Their orthogonality property is

$$\begin{aligned} \int_{-\phi_z}^{\phi_z} d\alpha \int_{\theta_z}^{\pi - \theta_z} d\theta \sin\theta Y_{\ell mk}^Z(\theta, \alpha, \theta_z, \phi_z) Y_{\ell' m' k'}^Z(\theta, \alpha, \theta_z, \phi_z) = \\ = \frac{4\phi_z \cos\theta_z \delta_{\ell\ell'} \delta_{mm'} \delta_{kk'}}{(2\ell+1)}, \end{aligned} \quad (24)$$

where θ_z defines the zonal subdivision of the angular half space in the azimuthal direction. For X-Y geometry θ_z is chosen to equal 0.

In the dependence on quadrant and angular zone the angle α is related to the angle ϕ used to express the angular dependence of the volume flux and source by the expressions:

$$\phi = \left\{ \begin{array}{l} \phi_n - \phi_{zn} + \alpha \\ 2\pi - \phi_n + \phi_{zn} - \alpha \\ \pi + \phi_n - \phi_{zn} + \alpha \\ \pi - \phi_n + \phi_{zn} - \alpha \\ \phi_n - \phi_{zn} - \alpha \\ 2\pi - \phi_n + \phi_{zn} + \alpha \\ \pi + \phi_n - \phi_{zn} - \alpha \\ \pi - \phi_n + \phi_{zn} + \alpha \end{array} \right\} \quad \text{for quadrants } X = \left\{ \begin{array}{l} E1, W1 \\ E2, W2 \\ E3, W3 \\ E4, W4 \\ N1, S1 \\ N2, S2 \\ N3, S3 \\ N4, S4 \end{array} \right\}, \quad (25)$$

where E, S, W and N denotes the east, south, west and north mesh side.

The spatial distributions of the angular flux and source moments inside of mesh (i,j) are now approximated by Legendre polynomial series:

$$\begin{pmatrix} \phi_{lmk}(\bar{r}) \\ q_{lmk}(\bar{r}) \\ S_{lmk}(\bar{r}) \end{pmatrix} = \sum_{\mu=0}^{K_1} (2\mu+1) P_{\mu}(u) \sum_{\nu=0}^{K_2} (2\nu+1) P_{\nu}(v) \begin{pmatrix} \phi_{lmk}^{\mu\nu} \\ q_{lmk}^{\mu\nu} \\ S_{lmk}^{\mu\nu} \end{pmatrix}, \quad (26)$$

where $K_1 = \text{Max}(0, M-(\ell+m)/2)$, $K_2 = \text{Max}(0, M-(\ell+m)/2-\mu)$,

and M is the order of the spatial flux and source approximation. The spatial moments $\phi_{lmk}^{\mu\nu}$, $q_{lmk}^{\mu\nu}$ and $S_{lmk}^{\mu\nu}$ are defined by

$$\begin{pmatrix} \phi_{lmk}^{\mu\nu} \\ q_{lmk}^{\mu\nu} \\ S_{lmk}^{\mu\nu} \end{pmatrix} = \frac{1}{4} \int_{-1}^1 du P_{\mu}(u) \int_{-1}^1 dv P_{\nu}(v) \begin{pmatrix} \phi_{lmk}(\bar{r}) \\ q_{lmk}(\bar{r}) \\ S_{lmk}(\bar{r}) \end{pmatrix}. \quad (27)$$

The coordinates u and v are defined by $x = x_{i-1} + \frac{1}{2}(x_i - x_{i-1})(u+1)$,

and $y = y_{j-1} + \frac{1}{2}(y_j - y_{j-1})(v+1)$, where x and y are the Cartesian coordinates.

The spatial dependence of the boundary angular fluxes (defined by Eq. (22)) on the mesh boundaries is similarly approximated by Legendre polynomial series:

$$\begin{pmatrix} \phi_{lmr}^{in} \\ \phi_{lmk}^{in} \\ \phi_{lmk}^{in} \end{pmatrix} = \sum_{\mu=0}^{K_3} (2\mu+1) P_{\mu}(w) \begin{pmatrix} \phi_{lmk}^{in\mu} \\ \phi_{lmk}^{in\mu} \\ \phi_{lmk}^{in\mu} \end{pmatrix}, \quad (28)$$

and

$$\begin{pmatrix} \phi_{lmk}^{in} \\ \phi_{lmk}^{in} \\ \phi_{lmk}^{in} \end{pmatrix} = \sum_{\nu=0}^{K_3} (2\nu+1) P_{\nu}(w') \begin{pmatrix} \phi_{lmk}^{in\nu} \\ \phi_{lmk}^{in\nu} \\ \phi_{lmk}^{in\nu} \end{pmatrix},$$

where $K_3 = \text{Max}(0, K-(\ell+m)/2)$, and K is the order of the spatial boundary flux approximation. The spatial moments are defined by

$$\begin{pmatrix} \phi_{lmk}^{in} \\ \phi_{lmk}^{in} \\ \phi_{lmk}^{in} \end{pmatrix} = \frac{1}{2} \int_{-1}^1 dw P_{\mu}(w) \begin{pmatrix} \phi_{lmk}^{in} \\ \phi_{lmk}^{in} \\ \phi_{lmk}^{in} \end{pmatrix}, \quad (29)$$

and

$$\begin{pmatrix} \phi_{lmk}^{Ei} \\ \phi_{lmk}^{Wi} \end{pmatrix} = \frac{1}{2} \int_{-1}^1 dw' P_v(w') \begin{pmatrix} \phi_{lmk}^{Eiv} \\ \phi_{lmk}^{Wiv} \end{pmatrix}.$$

The coordinates w and w' are defined by $x_B = x_{i-1} + \frac{1}{2}(x_i - x_{i-1})(w+1)$,
and $y_B = y_{j-1} + \frac{1}{2}(y_j - y_{j-1})(w'+1)$.

All computations with SURCU, FINELM and DIFF-2D (2DB) were performed at EIR on the CDC 6400 a machine which is ten to fourteen times slower than the CDC 7600. The times for a CDC 7600 were estimated from those for a CDC 6400 by dividing by ten in order to give a conservative estimate.

The SURCU calculations are summarized in Table 3. For all computations, the angular quadrants were not subdivided into angular zones ($I_z=1$). This approximation will be referred to as the quadruple QP_N .

Although the present theory is fully incorporated into the code SURCU, it has up to now only been tested for four angular quadrants, ie. for $I_z=1$. The effectiveness of the method for a greater number of angular intervals ($I_z > 1$) will be shown in future publications.

In Table 3, (N, K, M) indicate N th order surface flux approximation, Kth order spatial Legendre polynomial surface flux approximation along mesh boundaries and Mth order spatial Legendre approximations for the flux in the mesh.

Determination of the average flux is the most difficult part of the problem. The flux has its highest values in zone 1 where almost all fissions are concentrated. On the other hand zone 4, the second of the fission zones, has very low flux levels which, at least locally, at times can be negative for very coarse meshes and high order approximations. The order of the S_N and QP_N approximation is not very important and even low degree approximations lead to good results. However, the effect of the spatial flux approximation is very pronounced. It may be noted from Table 3 that the QP_0 approximation leads to poor results and the converged answer is far from the true values. However, already a QP_1 approximation leads to good results. With increasing degree of the Legendre polynomial for the spatial distribution of the flux the benchmark problem can be converged with fewer unknowns. This implies smaller demands on core memory and computational time for converging the problem.

In SURCU the user has the option of choosing the degree of the Legendre polynomial approximation in order to solve the problem most effectively. If the problem is materialwise too heterogeneous a high degree of polynomial leads to expensive calculations and the user is advised to use the linear approximation, $M=1$, which corresponds to differencing as used in TWODANT, TWOTRAN-II or DOT 3.5 and 4.

As is well known the flat flux approximation ($M=0$) is very inefficient and should be avoided. Best results are obtained when the order of spatial approximation within the mesh and on the boundary are identical ($M=K$).

CONCLUSIONS

The poor convergence of the finite difference schemes in TWODANT and DIFF-2D to the "exact" solution and the inability to converge even by using a very fine mesh shows the need to develop more accurate schemes, using higher order polynomials over coarse meshes, thus dramatically reducing the number of unknowns required. This reduces the demands on the computer storage and the computational time for a given convergence.

REFERENCES

- ¹ D. J. Dudziak, R. D. O'Dell, and R. E. Alcouffe, "Transport and Reactor Theory, July - September 30, 1981", Los Alamos National Laboratory report LA-9141-PR (January 1982).
- ² K. D. Lathrop and F. W. Brinkley, Jr., "TWOTRAN-II: An Interfaced Exportable Version of the TWOTRAN Code for Two-Dimensional Transport", Los Alamos Scientific Laboratory report LA-4848-MS (July 1973).
- ³ W. F. Walters and R. D. O'Dell, "Nodal Methods for Discrete-Ordinates Transport Problems in (x,y) Geometry", Proc. Topl. Mtg. Advances in Mathematical Methods for the Solution of Engineering Problems, Munich, FRG, April 27-29, 1981.
- ⁴ M. R. Wagner, "A Nodal Discrete-Ordinates Method for the Numerical Solution of the Multidimensional Transport Equation", Proc. ANS Topical Meeting on Comput. Methods in Nuclear Engineering, Williamsburg, VA, April 23-25, 1979, Vol. 2, p. 4-117.
- ⁵ J. Stepanek, "The DIN and QPN Surface Flux Integral Transport Method in One-Dimensional Geometries and X-Y Geometry", Proc. Internatl. Topl. Meeting, Munich, FRG, Vol. 1, (April 1981).
- ⁶ J. Stepanek, "The Multiple P_N Surface Flux Transport Method in X-Y Geometry Using General P_N Expansions of the Flux in Angle and Space", EIR report (to be published) (1983).
- ⁷ J. Stepanek, T. Auerbuch, W. Hälg, "Calculation of Four Thermal Reactor Benchmark Problems in X-Y Geometry", EIR report No. 464 (1982).
- ⁸ A. Kavenoky, M. Lam-Hime and Z. Stankovski, "Improvements of the Integral Transport Theory Method", ANS Meeting: "Computational Methods in Nuclear Engineering", April 23-25, 1979 Williamsburg.
- ⁹ A. Kavenoky, SERMA-CEN, Sacraiy, France (private communication) (1978)
- ¹⁰ Final Report of Co-ordinated Research Programme by the IAEA on "Transport Theory and Advanced Reactor Calculations", IAEA-TECDOC-254 (1981).

The convergence study for various M and K is to be found in Ref.7 .

Generally if an iterative scheme is not well balanced and properly accelerated the number of inner iterations increases very rapidly with increasing order of spatial flux approximation and increasing number of meshes.

For example, for DOT 3.5 (see Refs. 5 and 6), the problem could be converged only for a 4x4 mesh and a linear spatial flux approximation. For an 8x8 and higher number of meshes, the inner iterations could not be converged even after a large number of iterations. The DOT 3.5 results published in Ref.8 were later found to be not sufficiently converged.

As mentioned before the diffusion and the transport solutions of this benchmark problem are very close.

In order to compare the effectiveness of higher order polynomial approximations of the spatial flux as against the usual spatial differencing scheme and also draw a comparison between different polynomial approximations, such as nodal in TWOTRAN-NODAL or MULTIMEDIUM, Legendre polynomial in SURCU, two diffusion codes were used.

The first, FINELM is a 2-3-dimension finite element diffusion code which subdivides the spatial region into triangles and rectangles. The flux within an element is approximated by Lagrangian polynomials.

The second is DIFF-2D, a two dimensional finite difference diffusion code.

FINELM and DIFF-2D results are summarized in Tables 4 and 5 and in Fig.2 .

It is noticeable that with a finite difference scheme as used in DIFF-2D the "exact" solution cannot be obtained even with a 64x64 mesh net. A DIFF-2D solution with 128x128 meshes requires a memory larger than 400K octal and could not be performed on our CDC computer.

In contrast, however, FINELM produces a converged answer. Comparing the lower approximations in FINELM to DIFF-2D, one sees that FINELM results are better and require less computing time. Also, higher degrees of approximation in FINELM lead to better results and generally to shorter computational times than lower approximations.

FINELM delivers a good result using a 4x4 mesh and a higher order approximation in a very short computational time. Computations were performed with polynomial approximations up to degree four for rectangles and up to degree seven for triangles. Triangles usually offer a less accurate result than rectangles because they introduce an asymmetry into the rectangular X-Y geometry. The results with triangular elements have been introduced for comparison with rectangles. At the present time, approximations with rectangles are restricted to degrees up to four. However, higher orders will be introduced in the spring of 1983.

Comparing Lagrangian Nodal and Legendre Polynomial approximations of the same order, it may be seen that convergence is best for Legendre polynomials.

- ¹¹ D. Davierwalla, C. E. Higgs, and S. Pelloni, "FINELM - A Multigroup Finite Element Diffusion Code", Parts 1, 2, 3, EIR reports Nos. 419 (1981), 428 (1981) 459 (1982).
- ¹² R. Rühle, "RSYST - Ein integriertes Modulsystem mit Datenbasis zur automatischen Berechnung von Kernreaktoren", IKE-Stuttgart, Nr. 4-12 (1973) and
W. W. Little and R. W. Hardie, "2DB User's Manual", BNWL-831, Rev.1 (1969)
- ¹³ W. A. Rhoades and R. L. Childs, "An Updated Version of the DOT 4 One- and Two-Dimensional Neutron/Photon Transport Code", OakRidge National Laboratory report ORNL-5851 (July 1982)
- ¹⁴ H. Finnermann, F. Bennewitz, and M. R. Wagner, Atomkernenergie, 30, 123 (1977)
- ¹⁵ B. Müller and M. R. Wagner, "Improved Transport Theory Approximations for Heterogeneous Assemblies with Strong Local Absorbers", Tagungsbericht Jahrestagung Kerntechnik 1981, p. 55, Düsseldorf, FRG (March 1981).
- ¹⁶ M. R. Wagner and K. Koebke, "Progress in Nodal Reactor Analysis", Proceedings of TMI Meeting.

Table 1. LWR Pool Reactor, Eigenvalue, TWODANT and TWOTRAN-NODAL

Code	Number of Meshes	Average Fluxes ($\times 10^{-3}$)					k_{eff}	Total CPU Time (sec)	CPU Time per Mesh and Iteration (sec $\times 10^3$)
		Zone 1	Zone 2	Zone 3	Zone 4	Zone 5			
T	8x8	16.789	0.1589	0.1537	0.3722	0.9989	0.98277	33	1.2
W	16x16	16.849	0.1322	0.0492	0.3139	0.8405	1.00244	43	0.9
C	32x32	16.852	0.1259	0.0383	0.2973	0.7998	1.00758	46	1.0
D	48x48	16.859	0.1245	0.0369	0.2952	0.7931	1.00837	79	1.3
A	64x64	16.855	0.1244	0.0369	0.2946	0.7909	1.00858	100 ^a	1.2
N	80x80	16.853	0.1243	0.0368	0.2942	0.7898	1.00874	170 ^a	1.3
T	120x120	16.861	0.1241	0.0367	0.2939	0.7888	1.00886	440 ^a	1.2
T N	8x8	16.866	0.1542	0.0181	0.3597	0.9372	0.98662	276 ^b	2.6
W C	16x16	16.860	0.1309	0.0227	0.3097	0.8151	1.00413	294 ^b	2.6
C D	32x32	16.858	0.1249	0.0329	0.2952	0.7910	1.00841	490 ^b	2.5
T A	48x48	16.862	0.1242	0.0355	0.2941	0.7889	1.00880	1060 ^b	2.5
R L	64x64	16.860	0.1241	0.0362	0.2938	0.7883	1.00889	1760 ^b	2.6
A									
N									

a. CDC-7600 time estimated from CRAY-1 execution time

b. No iteration acceleration, very slow convergence

TWODANT convergence = 1×10^{-6}

TWOTRAN-NODAL convergence = 1×10^{-5}

Table 2. Nodal Discrete Ordinates Method 4 (NDCM), Code MULTIMEDIUM

Number of Meshes	Method Approximation	Average Fluxes ($\times 10^{-3}$)					k_{eff}	Total CPU Time (sec)	CPU Time per Mesh and Iteration (sec $\times 10^3$)
		Zone 1	Zone 2	Zone 3	Zone 4	Zone 5			
8x8	NDCM, NS ₃ (20)	16.822	0.1151	0.0891	0.2868	0.7704	1.01262	3.3	2.6
16x16	NS ₃ (12)	16.845	0.1215	0.0599	0.2901	0.7811	1.01023	7.8	1.7
24x24	NS ₃ (8)	16.853	0.1232	0.0448	0.2927	0.7859	1.00936	11.0	1.5
32x32	NS ₃ (6)	16.856	0.1238	0.0388	0.2938	0.7880	1.00898	18.6	1.1
48x48	NS ₃ (4)	16.859	0.1243	0.0343	0.2946	0.7893	1.00870	36.6	0.9
64x64	NS ₃ (4)	16.860	0.1245	0.0326	0.2948	0.7897	1.00862	67.7	1.0
64x64	NEM, ν_{2B}^2	16.863	0.1270	0.0274	0.3005	0.8019	1.00675	32.8	0.4

Computer used: CYBER 176

Pointwise convergence criterion $\epsilon = 0.5 \cdot 10^{-4}$

SP₂ approximation of surface fluxes on the node edges

Double Gauss quadrature used in 1-D discrete ordinate calculations

Flat, isotropic transverse leakage approximation

NS₃(k): 1-D S₃ calculations with k spatial intervals

NEM: Nodal expansion method 14 (diffusion theory), $D = 1/(3\Sigma_{tr})$ and $\Sigma_{tr} = \Sigma_t$

Table 3. LWR Pool Reactor, SURCU Calculations

Number of Meshes				Average Fluxes ($\times 10^{-3}$)					k_{eff}	Total CPU Time (sec)
	N	K	M	Zone 1	Zone 2	Zone 3	Zone 4	Zone 5		
4x4	0	0	0	16.408	0.5210	0.8624	1.0367	1.3320	0.77421	0.2
8x8				16.510	0.4864	0.6761	0.9736	1.4367	0.80459	0.6
16x16				16.718	0.3458	0.2935	0.6984	1.2922	0.87820	2.7
32x32				16.812	0.2132	0.1076	0.4382	0.9994	0.94773	8.1
4x4	1	0	0	16.433	0.5218	0.8160	1.0390	1.3308	0.77441	0.3
8x8				16.532	0.4827	0.6354	0.9703	1.4346	0.80600	1.3
16x16				16.728	0.3389	0.2750	0.6913	1.2882	0.88088	4.6
32x32				16.818	0.2212	0.1101	0.4603	1.0640	0.94803	13.5
64x64				16.845	0.1521	0.0601	0.3575	0.9067	0.98476	46.3
4x4	2	0	0	16.433	0.5218	0.8160	1.0390	1.3308	0.77441	0.5
8x8				16.532	0.4827	0.6355	0.9703	1.4347	0.80600	2.0
16x16				16.728	0.3388	0.2757	0.6913	1.2882	0.88089	5.5
32x32				16.817	0.2210	0.1111	0.4662	1.0640	0.94799	23.1
4x4	3	0	0	16.433	0.5218	0.8160	1.0390	1.3308	0.77441	0.7
8x8				16.532	0.4827	0.6355	0.9703	1.4347	0.80600	2.8
16x16				16.728	0.3388	0.2757	0.6913	1.2882	0.88089	7.1
32x32				16.817	0.2210	0.1111	0.4662	1.0640	0.94798	31.0
4x4	0	1	1	16.946	0.3291	-0.1260	0.6662	1.8768	0.88906	0.5
8x8				16.948	0.1569	-0.0130	0.3455	0.9108	0.99124	1.7
16x16				17.830	0.1159	-0.0035	0.2700	0.7820	1.01702	9.2
32x32				16.866	0.1053	0.0222	0.2463	0.6960	1.02501	44.4
64x64				16.865	0.0976	0.0231	0.2272	0.6532	1.03156	151.2
4x4	1	1	1	16.932	0.3342	-0.0999	0.6815	1.8523	0.88557	0.6
8x8				16.934	0.1672	-0.1037	0.3726	0.9543	0.98259	3.5
16x16				16.871	0.1303	0.0119	0.3063	0.8138	1.00480	15.4
32x32				16.858	0.1247	0.0351	0.2952	0.7912	1.00845	72.0
64x64				16.858	0.1241	0.0367	0.2941	0.7890	1.00862	201.3

Table 3(cont.). LWR Pool Reactor, SURCU Calculations

Number of Meshes	N	K	M	Average Fluxes ($\times 10^{-3}$)					k_{eff}	Total CPU Time (sec)
				Zone 1	Zone 2	Zone 3	Zone 4	Zone 5		
4x4	2	1	1	16.932	0.3342	-0.0999	0.6815	1.8523	0.88557	1.1
8x8				16.934	0.1672	-0.1033	0.3725	0.9544	0.98259	4.8
16x16				16.871	0.1303	0.0126	0.3062	0.8138	1.00480	21.8
32x32				16.858	0.1246	0.0362	0.2951	0.7911	1.00848	59.3
4x4	3	1	1	16.932	0.3342	-0.0999	0.6815	1.8523	0.88557	1.5
8x8				16.934	0.1672	-0.1033	0.3725	0.9544	0.98259	5.9
16x16				16.871	0.1303	0.0125	0.3062	0.8138	1.00480	29.6
32x32				16.858	0.1246	0.0362	0.2951	0.7911	1.00848	72.6
4x4	4	2	2	16.969	0.1457	-0.1678	0.3282		0.98851	0.8
8x8				16.871	0.1126	0.0125	0.2707		1.01072	3.4
16x16				16.861	0.1101	0.0307	0.2641	0.7451	1.01575	18.1
32x32				16.863	0.1061	0.0275	0.2512	0.7099	1.02200	91.1
4x4	1	2	2	16.959	0.1657	-0.1500	0.3616	0.8598	0.98858	1.9
8x8				16.872	0.1275	0.0113	0.3004	0.7979	1.00687	8.0
16x16				16.858	0.1243	0.0353	0.2944	0.7894	1.00871	37.1
32x32				16.858	0.1241	0.0365	0.2941	0.7890	1.00882	130.1
64x64				16.858	0.1240	0.0366	0.2940	0.7889	1.00888	303.1
4x4	2	2	2	16.959	0.1657	-0.1501	0.3615	0.8598	0.98859	4.1
8x8				16.872	0.1275	0.0105	0.3003	0.7977	1.00689	15.8
16x16				16.858	0.1243	0.0356	0.2943	0.7892	1.00874	64.8
32x32				16.858	0.1241	0.0370	0.2939	0.7886	1.00887	149.2
64x64				16.858	0.1240	0.0371	0.2938	0.7883	1.00890	391.2
4x4	3	2	2	16.959	0.1657	-0.1501	0.3615	0.8598	0.98859	6.6
8x8				16.872	0.1275	0.0105	0.3003	0.7977	1.00689	21.3
16x16				16.858	0.1243	0.0356	0.2943	0.7892	1.00874	78.9
32x32				16.858	0.1241	0.0370	0.2939	0.7886	1.00887	195.3
64x64				16.857	0.1240	0.0372	0.2937	0.7882	1.00890	561.1

Table 4. LWR Pool Reactor, FINELM and DIFF-2D Diffusion Calculations

Code	Number of Meshes	M	Average Fluxes ($\times 10^{-3}$)					k_{eff}	Total CPU Time (sec)
			Zone 1	Zone 2	Zone 3	Zone 4	Zone 5		
FINELM Rectangles	4x4	1	16.917	0.4619	-0.0726	1.0518	3.6762	.78158	0.3
	8x8		16.865	0.1852	0.0231	0.4336	1.1359	.96203	0.5
	16x16		16.866	0.1586	0.0222	0.3726	0.9439	.98299	1.5
	32x32		16.861	0.1500	0.0300	0.3517	0.8960	.98947	7.6
	64x64		16.859	0.1472	0.0339	0.3453	0.8820	.99151	52.2 ¹
	4x4	2	16.860	0.1564	0.0318	0.3653	0.9329	.98326	0.5
	8x8		16.864	0.1508	0.0255	0.3540	0.8961	.98892	1.6
	16x16		16.860	0.1474	0.0326	0.3452	0.8810	.99151	8.2
	32x32		16.859	0.1463	0.0352	0.3432	0.8775	.99217	57.2 ²
	4x4	3	16.872	0.1539	0.0110	0.3620	0.8765	.98867	1.4
	8x8		16.860	0.1473	0.0326	0.3451	0.8812	.99152	5.0
	16x16		16.858	0.1462	0.0353	0.3431	0.8774	.99220	34.5 ³
	4x4	4	16.858	0.1468	0.0368	0.3442	0.8931	.99104	2.3
	8x8		16.859	0.1463	0.0351	0.3432	0.8777	.99214	15.0
	16x16		16.858	0.1461	0.0356	0.3430	0.8770	.99226	75.6 ⁴
DIFF-2D	4x4	1	18.165	0.0486	0.0385	0.1145	0.3352	1.07753	0.5
	8x8		17.498	0.0893	0.0896	0.2090	0.5917	1.03965	0.9
	16x16		17.039	0.1206	0.0821	0.2837	0.7598	1.01221	2.5
	32x32		16.830	0.1364	0.0546	0.3215	0.8342	.99896	9.0
	64x64		16.757	0.1425	0.0414	0.3350	0.8601	.99417	35.4

M-means the order of the spatial flux approximation

1 2-x 2y dissectors

2 2-x 1y dissector

3 3-x dissectors

Diffusion calculations made with $D = 1/(3\Sigma_{tr})$ and $\Sigma_{tr} = \Sigma_t - \Sigma_{sl}/3$

Table 5. LWR Pool Reactor, Eigenvalue, FINELM Calculation *

	<u>M</u>	<u>Average Fluxes (x 10⁻³)</u>					<u>k_{eff}</u>	<u>Total CPU Time (sec)</u>
		<u>Zone 1</u>	<u>Zone 2</u>	<u>Zone 3</u>	<u>Zone 4</u>	<u>Zone 5</u>		
Triangles	1	16.572	0.4744	0.5620	0.8148	2.4167	0.82703	0.3
	2	17.340	0.1911	-0.8496	0.4369	1.0346	0.96508	0.5
	3	16.831	0.1551	0.0865	0.3625	0.8958	0.98705	1.2
	4	16.868	0.1474	0.0184	0.3456	0.8878	0.99109	2.4
	5	16.857	0.1470	0.0375	0.3440	0.8771	0.99194	5.8
	6	16.857	0.1462	0.0350	0.3431	0.8777	0.99219	14.7
	7	16.858	0.1461	0.0354	0.3430	0.8770	0.99225	39.7
Rectangles	1	16.917	0.4619	-0.0726	1.0518	3.6762	.78158	0.3
	2	16.860	0.1564	0.0318	0.3653	0.9329	.98326	0.5
	3	16.872	0.1539	0.0110	0.3620	0.8765	.98867	1.4
	4	16.858	0.1468	0.0368	0.3442	0.8931	.99104	2.3

M - denotes the spatial flux approximation

* 4x4 spatial flux mesh

Diffusion calculations made with $D = 1/(3\Sigma_{tr})$ and $\Sigma_{tr} = \Sigma_t - \Sigma_{sl}/3$

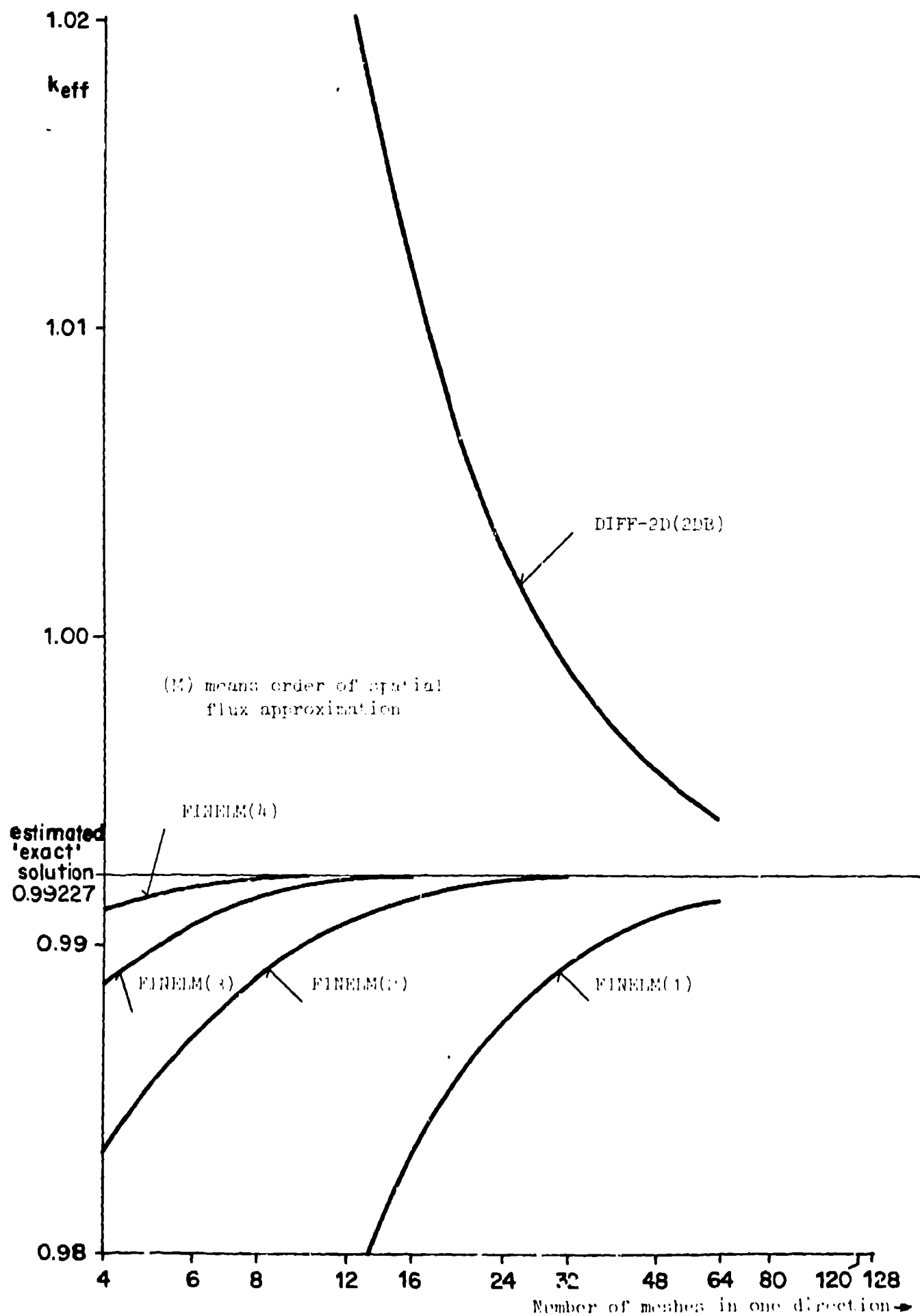


Figure 2. Eigenvalue as Function of Number of Meshes and Flux
Approximation for LWR Pool Reactor, Diffusion Calculations
made with $D = 1/(3\Sigma_{tr})$ and $\Sigma_{tr} = \Sigma_t - \Sigma_a/3$

